

Problem Set 1

Michelle Newcomer

Stat 242 Fall 2013

Problem 1

Class survey: done

Problem 2

a) I cloned the Git repository onto my machine using Github for Windows.

b) Commits made to my own repository on Git.

D:\Users\Michelle Newcomer\Documents\GitHub\LecturePractice [master]> git log commit c02baebc3597c4eeec4b80e78221bbdccde760fc Author: Michelle Newcomer mnewco8290@yahoo.com Date: Thu Sep 12 18:26:00 2013 -0700

Practice with Unit 3

commit 3380b7ad17fecf409aa93affb799d744346a6aa0 Author: Michelle Newcomer mnewco8290@yahoo.com Date: Mon Sep 9 09:15:58 2013 -0700

Practice with Unit 1 and Unit 2

commit 47eb8e3ee64583345b8ce38f319e89c84510689e Author: MNewcomer mnewco8290@yahoo.com Date: Fri Sep 6 10:22:44 2013 -0700

Lecture practice created

D:\Users\Michelle Newcomer\Documents\GitHub\LecturePractice [master]>

9/8/13 Commits · MNewcomer/Stat243PSets

This repository Search or type a command Explore Gist Blog Help MNewcomer

You don't have any verified emails. We recommend [verifying](#) at least one email.
Email verification helps our support team help you in case you have any email issues or lose your password.

MNewcomer / Stat243PSets Unwatch 1 Star 0 Fork 0

branch: master Stat243PSets / Commits

Sep 07, 2013	Tested Rmarkdown files MNewcomer authored a day ago	6996f38026 Browse code
Sep 06, 2013	Myls.sh changed again ... MNewcomer authored 2 days ago	33187acf2b Browse code
	Changed myls.sh ... MNewcomer authored 2 days ago	ffaec03405 Browse code
Sep 05, 2013	Added and deleted files ... MNewcomer authored 3 days ago	30830fc65b Browse code
Sep 04, 2013	Lecture Example myls.sh ... MNewcomer authored 4 days ago	89208c6ebf Browse code
Sep 03, 2013	Trial text document ... MNewcomer authored 5 days ago	dcf3b1ec3f Browse code
	Starting the GIT repository ... MNewcomer authored 5 days ago	4e3d1904a8 Browse code

© 2013 GitHub, Inc. Terms Privacy Security Contact Status API Training Shop Blog About

Problem 3

```

#!/bin/bash
cd ~/Documents/Stat242PSFiles/PS1_link/

echo "Please enter the code # of the type of crop then press enter: "
read input_variable
echo "You entered: $input_variable"
IFS=: # internal field separator
type=$input_variable # this specifies the type of crop

wget -O Data${type}.zip "http://data.un.org/Handlers/DownloadHandler.ashx?
DataFilter=itemCode:"${type}"&DataMartId=FAO&Format=csv&s=countryName:asc,elementCode:asc,year:desc&c=2,
#this line downloads the data and renames the data to Apricots.zip

#this for loop extracts each zip file and appends .csv to the original zip file name
for i in *.zip #this line counts the number of zip files
do
    n=$(unzip -lqq $i | awk '{print $NF}') #-l unzips the files names and -qq does it quietly while
    only giving the file names inside the zip to the screen. This then pipes the information to awk
    with searches the output to find the name of the fields
    e=${n#*.} # this line looks at the file name inside of the zip file and remove everything in
    front of the . and saves the csv portion
    unzip $i && mv $n ${i%.*}.${e} #this line unzips the file and renames the unzipped file to
    be the original portion plus csv
    rm $i #This then removes the original zip file because it has already been extracted
done

sed 's/,//g' *.csv > UN_No_Comma.csv #this removes the comma
sed 's/"/"/g' UN_No_Comma.csv > UN_No_Quote.csv #this removes the extra " in the country name
grep -i + UN_No_Quote.csv > UN_World_Regions.csv #this separates the regions from the countries
and save the regions
grep -i -v + UN_No_Quote.csv > UN_Countries.csv# this separates the countries and saves countries

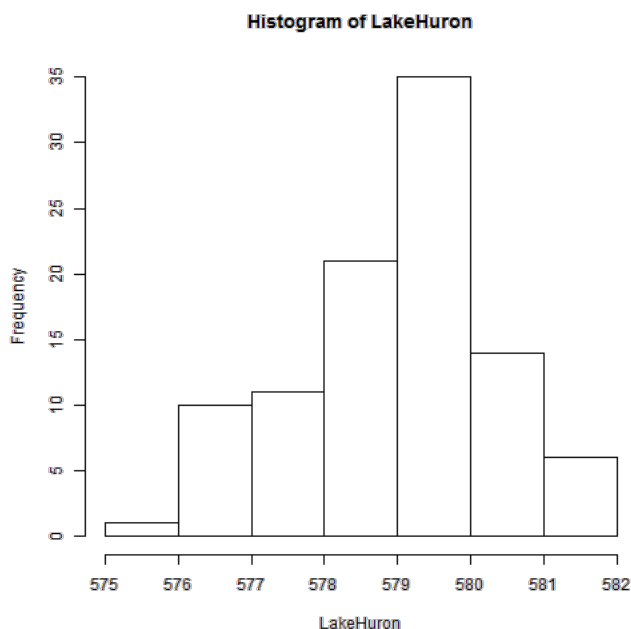
yrs=1965:1975:1985:1995:2005

for yr in $yrs
do
    grep -i -e ${yr} UN_Countries.csv > UN_Countries_${yr}a.csv #this grabs entries that have the
    year pattern
    grep -i -v ${yr}.[0-9] UN_Countries_${yr}a.csv > UN_Countries_${yr}b.csv #grabs the countries
    that do not have the pattern year. because that means it is a double and represents the acreage
    rm UN_Countries_${yr}a.csv
    grep -i -e Area -e Harvested UN_Countries_${yr}b.csv > UN_Countries_${yr}c.csv #grabs only
    the entries with words area and harvested
    rm UN_Countries_${yr}b.csv
    sort -t',' -k6 -r UN_Countries_${yr}c.csv | head -5 > ${yr}_top5.txt #sorts the 6th column,
    puts it in reverse order, then pipes the output to a text file
    rm UN_Countries_${yr}c.csv
done

```

Problem 4

```
hist(LakeHuron)
```



```
lowHi <- c(which.min(LakeHuron), which.max(LakeHuron))
yearExtrema <- attributes(LakeHuron)$tsp[1] - 1 + lowHi
```

Problem 5

This is an R Markdown document. Markdown is a simple formatting syntax for authoring web pages (click the **MD** toolbar button for help on Markdown).

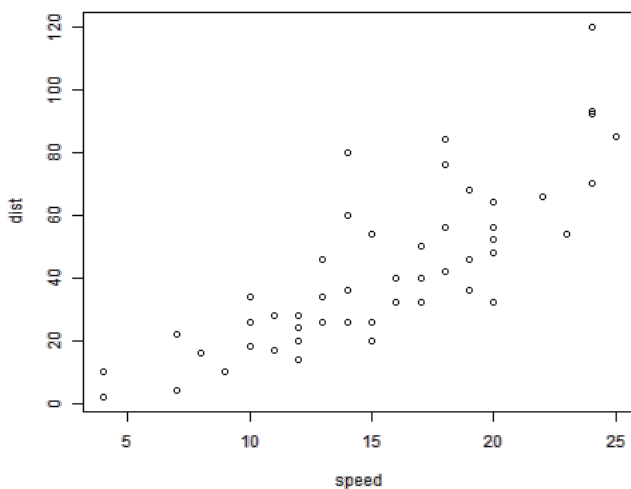
When you click the **Knit HTML** button a web page will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed      dist
##  Min.   : 4.0    Min.   :  2
## 1st Qu.:12.0    1st Qu.: 26
##  Median:15.0    Median: 36
##   Mean :15.4     Mean : 43
## 3rd Qu.:19.0    3rd Qu.: 56
##   Max. :25.0     Max. :120
```

You can also embed plots, for example:

```
plot(cars)
```



There are inline equations such as $y_i = \alpha + \beta x_i + e_i$

And displayed formulas:

$$\frac{1}{1 + \exp(-x)}$$